Word Recognition in Continuous Speech Using Linear Prediction Analysis

RICHARD WESLEY CHRISTIANSEN

UNIVERSITY OF UTAH

6

August 1976
UTEC-CSc-76-226
COMPUTER SCIENCE, UNIVERSITY OF UTAH
SALT LAKE CITY, UTAH 84112

(6) Word Recognition in Continuous Speech Using Linear Prediction Analysis,

by

(10) Richard Wesley/Christiansen

(9) Technical rept.,

(11) August 1976

(12) 89 p.

(14) UTEC-CSc-76-226

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract (15) No. DAHC15-73-C-0363, ARPA Order-2477

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

ABSTRACT

A promising method of automatic word recognition in continuous speech, recently designated as word spotting, has been demonstrated. The method uses error residual ratios from LPC (Linear Predictive Coding) vocoder analysis for waveform comparison and a dynamic programming procedure for time registration between the incoming speech and a template of the key word. Using a similarity threshold, the incoming speech is compared with several templates to account for variability in spectral shape. This system can work in real time using a real time vocoder.

The multiple templates are used in such a way that a small number of templates, ~~three or four~~, is made to look like several hundred or more. This is accomplished by dynamically constructing a composite template from parts of each single template as part of the processing of the incoming speech. ~~Thus,~~ a particular composite template is constructed for each word being recognized.

An accuracy of 99 ~~percent~~ with no false alarms was achieved using 205 key words, ~~five~~ different speakers, and approximately ~~ten~~ minutes of speech text. Performance in the presence of additive white gaussian noise of approximately 11 dB signal-to-noise ratio was 66 ~~percent~~. When the speech was processed to account for the noise, results improved to ~~85 percent to 90 percent~~ accuracy. Finally a digit

recognition experiment was performed using over 1200 digits spoken by ten different people with a resultant accuracy of 97 percent.

I. INTRODUCTION

With the rapid evolution of computers, automated computer control systems, the explosive expansion of knowledge, and the need to communicate rapidly and efficiently, there has developed considerable interest in automatic speech recognition. Some of this interest is directed toward automatic translation of speech into written text, voice-controlled dialing of telephones, voice control of manufacturing operations, automatic telephone orders or inquiries without a human intermediary, etc. Most approaches to this general problem involve rather complicated systems in which the acoustic waveform must be segmented and the segments then classified according to some set of tests and rules which are usually very complex.

A part of the general speech recognition problem is that of recognizing a single word. This problem has at least two levels of difficulty. The first and easier of the two is recognizing a single word spoken in isolation by a single speaker. Here, the system gets one or more training utterances, the word is well articulated, signal-to-noise ratio is large, the word is one of a small set of possible words and is spoken the same way each time. This problem becomes more difficult when any of these requirements are not met. Finally, if the word is part of some connected speech, we reach the second level of difficulty -- that of determining each time a specific word is uttered

in continuous speech. This has recently been designated as word spotting in continuous speech.

The problem described in this dissertation is word spotting in continuous speech. Here the objective is to process continuous speech using a digital computer, and to determine if and when a chosen reference word ever occurs.

This work has successfully applied easily calculated LPC parameters in a simple dynamic programming time warp algorithm developed by Bridle.[23] Multiple templates are then used in a way which is similar to the nearest neighbor method of pattern recognition. The most significant effect of the multiple template procedure is to allow a small number like three or four templates to be used in such a way that they look like a very large number, like several hundred or more. This is accomplished by dynamically constructing a composite template from parts of each of the single templates during the actual word spotting processing. This process is described in detail in section IV. This method may be utilized to recognize words spoken in isolation or as part of ordinary conversation.

The primary system evaluation was performed using approximately two-minute passages of well-articulated speech recorded in a quiet room by five different speakers and using multiple-syllable template words. The effect on performance of the system when the input speech was corrupted by additive gaussian white noise was examined. An attempt was made to extract the speech LPC parameters from the noisy speech using two methods described later. An attempt was also made

to recognize words in the noisy speech by adding similar noise to the templates. A preliminary evaluation of the system was made using multiple templates from different speakers. Finally, the performance of the system was evaluated by having ten different speakers each speak approximately one hundred digits chosen from a table of random numbers. For this final evaluation two templates of each digit spoken separately by speaker A were used for word spotting on speaker A's digit text. This was repeated for each of the remaining nine speakers. This was followed by a very brief experiment in which templates from one speaker were used to recognize digits spoken by another merely to give an indication of performance. The effect of noise on the digit recognition performance was not evaluated since the digit work was done only to suggest possibilities.

The overall accuracy was greater than 99 percent for multiple syllable words using clean speech with 100 percent accuracy for four speakers and 95 percent for the fifth. Using the single syllable word "Ford" from a sixth passage, 100 percent accuracy was achieved. For the cross-speaker experiments an overall accuracy of 95 percent was achieved. In all of the above cases, there were zero false alarms.

With an 11 dB signal-to-noise ratio, no processing to remove noise effects, and 66 false alarms per hour, the accuracy was 66 percent. This accuracy increased to 90 to 95 percent with processing to account for the noise. It was also found that the system does degrade smoothly in the presence of noise as shown in Fig. 8.

Finally, for the digit recognition experiments, an accuracy of 97 percent was achieved using five male and five female speakers.

No attempt was made to fully optimize the system even as it now is defined. In particular, the threshold, time-warp penalty factor, sample rate, and the number of poles in the LPC analysis were not optimized in any real sense. The values used, however, were not chosen completely arbitrarily. For example, Boll[31] indicates that a window width of 20 to 30 ms will yield acceptable quality synthetic speech for a 12-pole LPC vocoder. Since data are stored on the magnetic disc in blocks of 256 samples, a window width of 25.6 ms was a convenient size to use for the initial work. Then, since it worked so well initially, the 25.6 ms window was used for the remainder of the work.

Major problems with the system which require additional research are the need to readjust the threshold with each speaker and word and the poor cross-speaker performance (95 percent). Also, since the system uses a simple spectral comparison, it will probably have a higher false alarm rate on words that sound alike and on short words. This possible problem, however, was not examined in detail and no attempt was made to include similar sounding words in the various passages used. But, if one wanted to spot longer words, pairs of words like full names of individuals or longer phrases of words, the system should work quite well.

It is recognized that the system developed for this research is not a final complete system for word spotting, that it does not

meet all the practical requirements for such a system, nor does it address all the difficulties inherent in solving such a problem. In fact the developed system is less than optimum in performance in that it uses only a single feature to measure similarity between the reference word and the incoming speech. The main objective was to determine the best performance that could be achieved if we were allowed to tune the system for best performance for each word and each speaker. Even the concept "best performance" is somewhat vague in that there may be a trade-off between false alarm rate and number of misses of the correct word. Thus, "best performance" may be a function of the application. It is expected that if additional features such as pitch, voicing, zero crossing rate, etc., are included in the similarity function, the performance should improve and become more speaker independent.

A key point that should be made here is that this word spotting system has much more narrow and restricted objectives and applications than the more general word and speech recognition work being pursued elsewhere. No claims are made with respect to its usefulness in the speech recognition and speech understanding areas. It has been demonstrated that Bridle's dynamic programming algorithm does work quite well and that it, along with LPC analysis, does appear to form a sound basis for further research.

In summary, a system has been demonstrated, using only a single feature for comparing waveforms, which shows great promise for use in word spotting applications. Even with a false alarm rate of 66 per

hour, a reduction in listening time of approximately 100 to 1 is realized if we assume an average speaking rate of two words per second. For this case, the speech would be processed by the system, the spotted words along with their location on the tape would be stored, and then later listened to by an operator who would reject the false alarms. If the desired key words were present, the operator would then go back to the original tape and listen to the appropriate text in real time.

II. LITERATURE REVIEW

Introduction

During the last 25 years, there has been considerable interest and effort directed at determining what features of human speech can be used to recognize words and connected speech by using some form of automatic analog or digital signal processing system. Many attempts have been made to develop phonetic typewriters, automatic telephone dialers, isolated word recognizers, and speech recognition or understanding systems. A given system will typically use one or more features such as: formants, band-pass filter outputs, signal energy, pitch, sound spectrum, voicing, zero crossing rate, LPC parameters, etc.

If the word to be recognized is part of some continuous speech, then the speech signal is usually segmented into some basic recognizable parts such as phonemes or syllables. These elements are then classified and semantic and syntactic rules used to make decisions as to meaning.

One problem which makes these procedures difficult is that the acoustic or frequency features of a given phoneme or syllable vary with context, location in the sentence, and stress, even when spoken by the same speaker.[13] Time normalization is also a problem since speakers

speak at different rates, depending on a variety of conditions.

As the existing literature is reviewed, it will be found that each of these problems is addressed by one or more of the researchers, but that no really all-encompassing satisfactory solution has been found. It will also be noted that the most recent work uses extremely complex and often *ad hoc* analysis procedures. Although there has been, and currently is, a considerable amount of effort being expended, the general speech recognition problem still has not been solved.

One objective of the following literature review is to describe those current and previous research and development efforts which address directly or indirectly problems, techniques, and technology useful for solving the problem of word spotting in continuous speech. Another objective is to define the current capability and reported performance of word-spotting systems or word-recognition systems which are adaptable to the word-spotting problem.

In the discussion of the various speech and word-recognition papers which follows, no attempt is made to discuss all research and development efforts described in the literature. Instead a brief description of work which illustrates how the research has evolved is given in chronological order. These sources are given in the list of references. A more complete list of work reported will be given in a bibliography following the references.

Previous Work

The first limited vocabulary, automatic word recognizer was

developed at Bell Laboratories and reported in 1952 by Davis, Biddulph, and Balashek.[1] Their device could recognize the ten digits spoken over the telephone with 97 to 99 percent accuracy. It utilized 100 training utterances for each word and had to be trained for each speaker using the system. It required at least a 350-ms pause between each word. For analysis it calculated the first and second formants and formed a two-dimensional plot of formant 1 versus formant 2. A pattern-matching network then compared the unknown or input pattern with the set of ten stored reference patterns. The decision function then decided which digit yielded the best match to the unknown pattern. Best match was determined by calculating a correlation coefficient between the unknown and each reference and choosing that with the highest coefficient. This system was very speaker-dependent.

In 1956 Wiren and Stubbs[2] at Northeastern University produced a device which may be described as a successive binary classification scheme. The nature of this technique placed stringent requirements on each step in the process since overall accuracy was the product of the accuracy of each block in the series. This system performed a phoneme classification of spoken English. Step 1 used the output of a filter in the range of pitch frequency and a threshold to make a voiced versus unvoiced decision. Step 2 then used the output of a band-pass filter in the range of the first formant and a threshold to make a voiced turbulent versus voiced nonturbulent versus unvoiced turbulent decision. These signals were then successively classified as nonturbulent

vowel or vowel-like. If they were vowel sounds, then they were decoded into acute versus grave. Voiced turbulent sounds were decoded into stops or fricatives, and unvoiced sounds into the remaining stops and fricatives. Finally, diffuse, compact, and nasal characteristics were brought to bear to complete the classification.

This approach did not address the difficulties created by the fact that some sounds differ appreciably in their acoustic properties depending on whether they were spoken in context, in isolation, in the middle, at the beginning, or at the end of an utterance. The accuracy of various elements of this system was given as varying from 86 percent to 98 percent. No overall accuracy figure was given.

In 1956 and 1961, Olson and Belar at RCA reported on a phonetic typewriter development which recognized 7- or 10-word vocabularies in the earlier version[3] and a 100-word vocabulary in the later version.[4] The system calculated a three-dimensional sound spectrum with dimensions of intensity, time, and frequency. The first version used two levels for the intensity, while the later version used eight levels. Both systems used five time intervals and eight band-pass filters. A reference matrix was stored for each syllable formed by averaging 100 utterances of the syllable. An input or unknown syllable matrix was then examined for the presence of certain nonzero elements. If these elements were nonzero, a relay closed and the syllable was typed. Two problems with this system were that the syllables had to be enunciated very clearly, and they had to be spoken

in isolation. The system was speaker-dependent and had to be retrained for each speaker. The claimed accuracy was 98 percent.

In 1958 Dudley and Balashek[5,6] reported on a follow-up to the 1952 Bell Laboratories digit recognizer. This machine was similar to the earlier one, but instead of the first two formants, it used ten band-pass filters and calculated the correlation between the unknown and the reference based on energy in each frequency band. It also compared the duration of the phonetic pattern as in the earlier version. This system frequently achieved 100 percent accuracy for a single speaker. Again, the speaker had to say the words with about 350 ms isolation, 300-700 ms duration, and speak the same way as during the 100 training utterances. This system, as before, recognized the ten digits.

In 1959 Forgie and Forgie at Lincoln Laboratories[7] reported on a system which could recognize ten English vowels spoken in isolated words. This system derived pitch and first, second, and third formants from a 35-channel filter bank. To accomplish this, the outputs were envelope-detected, sampled, and processed on a computer. The filter bank covered frequencies from 115-10,000 Hz, and the output was sampled 180 times per second. The vowels were identified primarily by the location and duration of points in a plot of formant 1 versus formant 2. This sytem was tested using eleven male and ten female speakers and yielded an accuracy of 88 percent. When the duration information was used, accuracy increased to 98 percent. The isolated words were

of the form |b| - vowel - |t|. A key element introduced in this work was the use of a digital computer to process the sampled analog data.

In 1960 Denes at University College, London, and Mathews at Bell Laboratories[8] reported on a word recognizer which used a digital computer to perform time-frequency pattern matching. In this system the speech signal was applied to a 17-channel filter bank covering the range 200 to 4,000 Hz. The channel outputs were sampled sequentially 70 times per second. Time normalization was performed by linearly stretching all utterances to a standard length. The system recognized the ten digits spoken in isolation. Recognition was accomplished by cross-correlating the unknown processed data with the reference arrays of data and choosing the digit with the maximum correlation. This system was able to get zero errors with one particular speaker. Using reference patterns made by averaging over five different speakers, an error rate of 6 percent was achieved.

In 1967 Reddy, at Stanford University,[9] reported on a digital computer system for automatic recognition of connected speech utterances of one to two seconds length. In this system the speech signal was input directly to computer magnetic disc storage through an analog-to-digital converter. The 2-s utterances were first divided into sustained and transitional segments, using intensity and zero-crossing measurements in the decision function. A pitch period was then identified for each voiced segment using data near the center of the segment and analysis similar to that of Gold.[10] The amplitudes of the first 100 harmonics of the fundamental pitch frequency were then determined

from a Fourier expansion using data from one pitch period near the center of the voiced segment. For unvoiced segments, the spectral energy distribution was determined by performing a Fourier expansion using a time window near the segment's center. The window was about the length of an average pitch period for the speaker (approximately 10 ms). Using the results of this data analysis, 21 parameters, related to amplitude and frequency of formants, noise, pitch, zero crossings, and intensity, were compiled for each segment. Using these parameters, the segments were then classified as vowel-like, fricative-like, etc., with transition segments being rejected. These segments were then further classified as standard English phonemes. For a single speaker speaking 287 phonemes, 81 percent accuracy was reported. Computation time was 40 times real time. This procedure used many *ad hoc* procedures, both in classification and selection of analysis points.

In 1968 Purton at British Telecommunications Research, Ltd., England,[11] reported on a digit recognizer based on autocorrelation analysis followed by pattern matching. In this system, using a digital computer, the incoming speech was filtered into two signals, high frequency and low frequency bands, using filters with a crossover frequency of 1000 Hz. Both signals were then infinitely clipped and the resultant signals fed to two separate autocorrelators. The two autocorrelator outputs were then coded into a single 36-bit output which was sampled every 25 ms. Time warping was accomplished by stretching or contracting the autocorrelation of each word to be 30 samples by repeating or eliminating samples at appropriate intervals. Master or template

autocorrelation patterns were constructed using approximately five utterances of each word spoken by the same speaker. Recognition was then accomplished by comparing an unknown 30-sample word with the templates, using an *ad hoc* pattern matching and scoring procedure. For the ten digits, the claimed accuracy was 90 percent. The system was quite speaker-dependent, yielding 59 percent accuracy in a cross-speaker test.

Shearme and Leach at the Joint Speech Research Unit, Middlesex, England, reported in 1968[12] on some experiments in isolated word recognition using a 32-word vocabulary. Their system used envelope detectors on the outputs of 20 band-pass filters and sampled the outputs every 10 ms. Word matching was accomplished by defining a distance measure between the unknown word and the template. Distance was defined as the sum of the absolute values of the differences between corresponding ordinates in the 20-dimensional space. The decision function was to choose the word which yielded the shortest distance. The system allowed the use of multiple templates. The percent correct recognition increased from approximately 58 percent for one template to approximately 90 percent when nine templates were used in an experiment which was mainly a cross-speaker experiment. True time warping was not done.

In 1974 Gillman[14] and Weeks[15] at System Development Corporation reported on a linguistic analysis system used to recognize 160 words. This system performed an acoustic-phonetic analysis to calculate pitch, energy, formants, and several other parameters for 10-ms windowed speech samples. The system performed a linguistic analysis, utilizing a predictive parser to predict all possible words (from its limited

vocabulary) that could begin an utterance. It then predicted a list of words possible syntactically following the best matching first word and so on through the utterance. If at any step all words in the predicted list failed to match, then it backtracked to a second-choice word found earlier. A 30-rule grammer was used in the linguistic analysis. In a trial run of 20 utterances, the first place word was correct 89 percent of the time. Most errors were due to coarticulation effects between words.

Gillman also reported in 1974[16] on a system to distinguish between the three nasal phonemes, |m|, |n|, and |ŋ|. In this case the speech was low pass filtered at 9 kHz, sampled at 20 kHz with a 12-bit A/D. Windowed segments of speech 25.6 ms long were used to calculate 24 LPC coefficients every 10 ms to derive the first and third formant frequencies. A modified Euclidian distance function was used for pattern matching between the unknown speech and a table of nasal formant frequencies. This system required that end point detection for each nasal phoneme be done in advance of the analysis. The system was very speaker dependent and achieved an accuracy of 72 percent.

Also, in 1974, Molho at System Development Corporation[17] reported on an experimental interactive system for automatic recognition of fricatives and plosives in continuous speech. His system computed eight linear predictive coefficients in 10 ms intervals. It first scanned the speech samples over each 10 ms interval to pick peak

magnitude. The LPC analysis window was then centered at the half peak magnitude point. Following this analysis a spectrum test was performed to eliminate both strongly voiced and silent segments prior to classification. The classifier then assigned articulation labels to each segment with a score for ranking assigned to each label. It then located spectral peaks and determined the frequency, amplitude, amplitude rank, and a measure of sharpness for each peak. Using this information it then computed 23 binary test functions and employed these in a matrix classification scheme. Accuracy varied between 57 percent and 88 percent, depending on the type of sound.

In 1975 Schwartz and Makhoul at Bolt Beranek and Newman[18] reported on an acoustic phonetic recognition program that was part of a BBN Speech Understanding System. This system computed parameters for 100 frames per second. For each frame an FFT was computed on 20 ms of the windowed signal. Parameters computed were energy, 14-pole selective linear prediction[20] coefficients, formants, and a set of "slow" and "difference" parameters which reflect the time-varying behavior of some of the other parameters. The speech was then segmented with confidence numbers associated with various segmentation options. Classification or labeling of segments was accomplished by comparing average values of the parameters to thresholds for several features. Confidence numbers were then associated with the segment labels. The percentage of labeling errors for five male speakers speaking fifteen sentences was reported as 21 percent.

In 1967 Teacher, Kellett, and Focht at Philco Ford[21] reported on an experimental, limited vocabulary speech recognizer. This system was tested on the ten digits spoken in isolation. Only three parameters were used. The parameters were: single equivalent formant frequency, single equivalent formant (SEF) amplitude, and the state of voicing. The SEF frequency was obtained by measuring the elapsed time between the glottal impulse and the next following zero crossing. The SEF amplitude was defined to be the highest peak during that interval. Features required for each word in the vocabulary of the system were hard wired on a separate card for each word. A sequence detector was used to detect the presence of the required features from the unknown word in six sequential coupled stages in cascade. Feedback was employed to make the sequence detection independent of the rapidity with which the word was spoken and of varying speed within the word. Reported accuracy was 90 percent correct, 1 percent error, and 9 percent no response.

In December 1974, Bridle and Brown at the Joint Speech Research Unit, Middlesex, England,[23] reported on an experimental automatic word-recognition system which used continuous speech as input. Their system used a 19-channel vocoder which provided as output a 19-point logarithmic, short-term power spectrum with a nonlinear frequency scale. The vocoder produced its output at the rate of 50 frames per second and also a voiced-unvoiced decision each frame. A cosine transformation was applied to this output and the resultants were weighted by some empirically

derived coefficients. Following this, a modified Euclidian distance between a frame of input speech and a frame of a template word was formed. This distance was then transformed into a frame similarity function which varied between zero and one. Finally the resultant frame similarity number was processed through a time-warping algorithm which used a similarity threshold, a time-warp penalty factor, and a weighting coefficient as parameters in a novel dynamic programming scheme. Using a single speaker and with word templates extracted from the continuous speech text, recognition accuracies of 85 percent, with false alarms at the rate of 6 per hour, were reported.

Itakura at Bell Laboratories reported in 1975[22] on an isolated word-recognition system utilizing the minimum prediction residual of linear predictive coding analysis and a dynamic programming time-warping algorithm. This system used the minimum prediction residual as a measure of similarity between a frame of input speech and a frame of a word template. For preprocessing, first the speech signal was low-pass filtered and digitized, word end points were detected, and a long time spectrum normalization was performed. An 8-pole LPC calculation was performed using 30 ms windows with each window overlapping the previous window by 50 percent. Word matching was performed by choosing that word which had the minimum "distance" between itself and the unknown word. Distance was defined as the total log prediction residual of the input signal which was minimized by optimally registering the template LPC onto the input autocorrelation coefficients using a dynamic

programming algorithm. Reported recognition accuracy for a single speaker talking through the telephone system was 97.3 percent. Analysis required approximately 22 times real time.

In 1976 Rabiner and Sambur at Bell Laboratories[28] reported on some preliminary experiments in recognizing connected digits using a speaker independent system. Although their system worked on strings of connected digits, it did require that the number of digits in the string be provided as an input to the system. Their system was comprised of two parts. The function of the first part was to segment the digit string into individual digits. To accomplish this, a voiced-unvoiced-silence contour was constructed using LPC parameters, zero crossings, autocorrelation coefficients, and energy. Statistical information about the contour and the speech energy measurements was then used to perform the segmentation into single digits. The second part of the system then used the segmentation information to recognize the digits. For recognition purposes, four acoustic parameters, giving a gross measure of the nature of each phoneme, were used to classify the sounds in terms of six broad speech categories. These speech categories were then used to classify the digits. Zero-crossing rate statistics were used to perform a "self-normalization" and thus avoid using fixed threshold levels in the decision process.

The system required several minutes analysis for a string of three digits. Recognition accuracy was reported to be 91 percent using high quality soundproof booth recordings and 87 percent using recordings made in a noisy computer room.

Conclusion

From reading the above summaries, it is clear that most of the work reported to date requires complex acoustic and linguistic analysis and the use of many *ad hoc* and empirical procedures, rules, and thresholds to yield reasonable success.

One might correctly argue that a fully automatic word recognizer which can successfully compete with the human brain will not be achievable until we fully and correctly understand how the ear-brain system processes speech. One can also argue with even greater certainty that this understanding, if achievable at all, will not be achieved for several decades. J. R. Pierce[32] in 1969 challenged the entire community of speech recognition workers in an article in which he criticized the aims, methods, and scope of work in this field. He further maintained that a meaningful solution to the general recognition problem is beyond our present capacity.

In spite of the above criticisms, I believe that meaningful progress can be achieved in finding solutions to more specific and limited objective word recognition problems such as the word spotting problem described in this dissertation. The approach to be taken is the engineering approach referred to by G. Fant[33] in his book on speech sounds and features. The major advantage of the engineering approach is the elimination of phonetic decisions and the need to segment, classify, etc. The approach is therefore simple.

A major disadvantage is that speaker-dependent elements may dominate the similarity measure while the greatest problem, according

to Fant,[33] is to achieve a proper time normalization.

It is believed that a major step toward solving the time normalization problem has been achieved in the dynamic programming procedures utilized in the recent work of Itakura[22] and Bridle.[23] Of all the literature surveyed, only that of Bridle and Brown directly addresses the problem of word spotting in continuous speech. Some work that might be easily applied to the word-spotting problem which uses a potentially fast and conceptually simple algorithm with a minimum of *ad hoc* steps is the recent work reported by Itakura,[22] Boll,[24] and Coker.[25]

In the following section a different method of word spotting will be described. This method uses a linear prediction residual ratio for waveform comparison, the dynamic programming algorithm of Bridle for time normalization, and multiple templates to account for speaker variability. The templates are used in a frame-by-frame nearest-neighbor comparison scheme which has the effect of dynamically creating a composite template from the multiple templates as each frame of incoming speech is processed. The details of this procedure are described in Section IV.

III. RESEARCH DESCRIPTION

Approach

From the literature review summary, one can see that there are already rather extensive efforts underway in the general area of automatic speech recognition, speech understanding, and word recognition. In these previous and current investigations, various approaches to the problem of preprocessing to derive an initial set of features have been used. Features which have been used include outputs of bandpass, low-pass, and high-pass filters, formant frequencies and amplitudes, pitch, energy, voicing, zero-crossing rate, LPC coefficients, other LPC parameters, etc. The selection and derivation of secondary features and the design of classification algorithms has been largely *ad hoc* and empirical. Of all the work reviewed, only one researcher has directly pursued the problem of word spotting in continuous speech as an end in itself.

Although a solution to the word spotting problem might be derived from the more general work in speech recognition, that work has more ambitious objectives and as a consequence utilizes much more elaborate and complex procedures than are required for the simpler word spotting problem. In fact, the complexity of this problem probably lies somewhere between the two extremes of isolated word recognition and continuous speech recognition.

In searching for a solution, it was decided to keep it as simple as possible and avoid the problems of segmentation, classification, semantics, syntax, and complicated rules if at all possible. To accomplish this, one could design a system using a matched filter based upon some average representative template of the word to be spotted.

However, even with a representative template, time registration would still be a problem. If a template matching approach were to be used, then it would probably be desirable to construct the composite template dynamically as the incoming speech is processed to account for the varying speeds of speaking, pitch inflection, variable articulation effects, etc.

Bridle's interim report[23] described a system which did use a template matching approach and performed a nonlinear time registration of the template to the incoming speech. Bridle's system was simple and used a threshold to decide when the "distance" between the input speech and the template was close enough for the word to be present in the continuous speech. His system also used a very simple dynamic programming algorithm for nonlinear time warping, operated on continuous speech by analyzing one frame at a time, and did not require very much computer memory. Furthermore, any "frame similarity function" bounded by zero and plus one can be used without modifying his algorithm. The details of Bridle's algorithm are described in Section IV.

For primary parameters, Bridle used the amplitudes of the nineteen channels and a voicing indication from a channel vocoder. These parameters were transformed and scaled with an empirically derived

scale factor and then used to construct a frame similarity function. This simple procedure worked almost as well as some of the earlier isolated word recognizers which used filter banks.

It has also been shown[22,24] that certain ratios of linear prediction residuals are equivalent to the integrated ratio of the inverse filter spectra determined by the linear prediction coefficients. In fact, Itakura has shown[22] that one can achieve greater than 97 percent accuracy in isolated word recognition using the minimum prediction residual ratio as a measure of speech waveform similarity. Boll has also shown[24] that certain ratios of linear prediction residuals are bounded by zero and plus one. The specific residual ratios used and the details of the procedure are described in Section IV.

Based on the above, it was decided to utilize a ratio of linear prediction residuals as a frame similarity function directly in Bridle's dynamic programming algorithm and augment that system with a novel multiple template analysis scheme.

In summary, the main features of this approach are the following:

1. The approach is based on waveform template matching and does not require waveform segmentation, syntactic information, or semantic information.
2. The basic waveform features are linear prediction coefficients and autocorrelation coefficients.
3. The measure of similarity between a template frame and a frame of incoming speech is the minimum prediction residual.

4. Multiple templates are used to improve performance and to allow word spotting using templates and speech from different individuals.

The mathematics and details of this approach are presented in the next section.

Research Objectives

There were five main objectives of this research.

1. Given a recording of someone talking, i.e., a continuous speech recording, and a reference example of the same person speaking a word which occurs in the continuous speech text, we want to recognize the reference word every time it occurs in the recording using digital signal processing techniques.
2. Evaluate the system performance using "Quiet Room" quality recordings.
3. Corrupt the speech from No. 2 above with additive "white gaussian noise" and determine at what signal-to-noise ratio the system begins to degrade in performance.
4. Use three procedures in an attempt to improve performance using the noisy speech from No. 3 above. The three procedures are described in section V.
5. Provide a preliminary evaluation of system performance using multiple templates from multiple speakers other than the continuous text speaker. This is referred to as cross-speaker word spotting.

Contribution of this Research

The major contributions of this work can be summarized as follows:

1. It applied LPC analysis for waveform matching to the problem of word spotting in continuous speech.
2. It combined the dynamic programming time-warp procedure of Bridle with the inverse of the linear prediction residual ratio described by Boll and Itakura as a similarity measure.
3. It utilized a novel procedure for dynamically combining multiple templates into a composite template adapted to each word being searched for in the continuous speech.
4. Performance of the system using speech corrupted by additive white gaussian noise was examined. Three very different but very simple procedures for extracting useful LPC parameters from the noisy signal were tried. (Suggested approached for further research in this area are given in section VI.)
5. Performance of the system using noisy speech improved greatly if noise with the same statistics was added to the templates.
6. The performance of this system for word spotting is significantly improved over that of Bridle and Brown and can be done in real time using a real-time LPC vocoder.

The system developed here is not a final system but does indicate promising directions for future research.

IV. SYSTEM DESCRIPTION

Introduction

A detailed description of the word spotting procedure and algorithm will be given, along with pertinent mathematical relations. Basically this system has the effect of examining the incoming speech one frame at a time, performing a time registration (which has the effect of either compressing or stretching the incoming speech) as it slides the incoming speech past the template. It then makes a test to determine if the speech signal aligned with the template matches the word.

This process will be explained in four parts. In the first part, the construction of the template features will be discussed. This is followed by a discussion of the processing of the incoming speech and the calculation of the frame similarity function which provides a similarity measure between a frame of the template and a frame of incoming speech. The use of the frame similarity function in the time-warp algorithm will then be discussed, followed by a description of how multiple templates are used to improve performance, versatility, and robustness of the system.

Both the templates and the speech text receive the same front-end processing. That is, all speech is low-pass filtered at 4 kHz

and sampled at 10 kHz using a 14-bit A/D converter. The samples are then stored on magnetic disc for future processing. This processing is shown in block diagram form in Fig. 1.

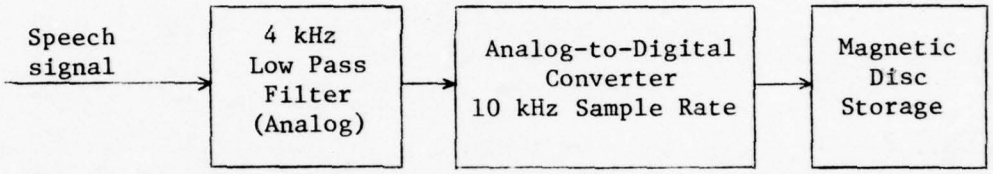


Fig. 1. Preprocessing block diagram.

Previous LPC vocoder work performed by Boll and others indicates that a 12-pole predictor with a 10 kHz sample rate using a Hamming window with a width of 20 to 30 ms provides synthetic speech of satisfactory quality. However, these values have not been optimized with respect to the problem of word recognition. Optimization of these parameters might very well show that fewer poles and a different window width would yield improved performance and/or shorter analysis time. The selection of these parameters is a subject which might warrant further work in the future for the development of a more efficient working system.

Template Construction

A reference template is first created by having an individual speak the word in isolation or by extracting the word from some

continuous text. The end points of the template are then determined so that the template is an integer number of frames long. This is done by displaying the template waveform on a screen, visually selecting the beginning and ending frame, and then playing back the portion selected for audio verification. Each frame is 256 sample points or 25.6 ms of speech.

Analysis proceeds by using data from each frame in sequence. The samples are first scaled using a Hamming window function. Thirteen autocorrelation coefficients, R_k, are then calculated and normalized by dividing each one by R_0. This has the effect of applying an automatic gain control on the speech.

The following description of linear prediction is taken from Makhoul.[26] The all-pole model assumes that a signal sample S(n) is derivable from a linear combination of past values and some input u(n) as follows:

$$S(n) = -\sum_{k=1}^{p} a(k)\, S(n-k) + Gu(n)$$

where G is a gain factor. In the case of speech production, the driving term, u(n), is unknown. Therefore, the signal, S(n), is only approximately predicted by the previous p samples. Thus we get an estimate of S(n) which we call $\hat{S}(n)$, where

$$\hat{S}(n) = -\sum_{k=1}^{p} a(k)\, S(n-k)$$

The error between the true value, S(n), and the predicted value, $\hat{S}(n)$, is given by:

$$e(n) = S(n) - \hat{S}(n) = S(n) + \sum_{k=1}^{p} a(k)\, S(n - k)$$

Then, rearranging the equation, we get the discrete all-pole linear prediction model for a short segment of speech, which is:

$$S(n) = -\sum_{k=1}^{p} a(k)\, S(n - k) + e(n) \quad n = 0, 1, 2 \ldots N - 1 \qquad (1)$$

where

N = number of samples in the window

a(k) = predictor coefficient

S(n) = nth speech sample

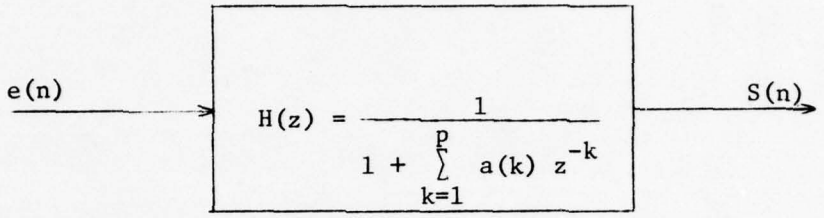
e(n) = nth error term

p = number of poles in the LPC model

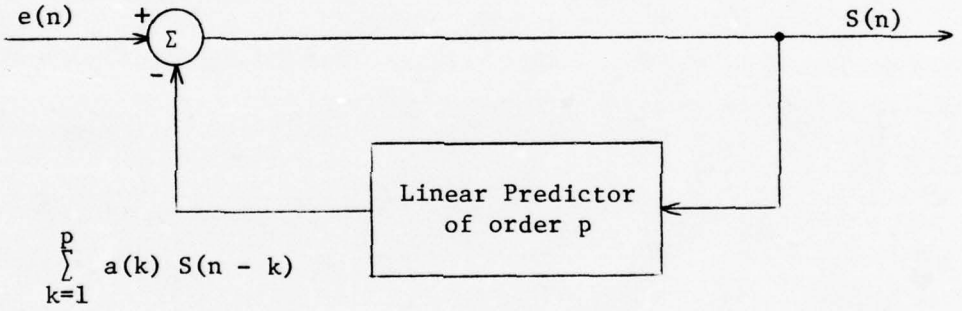
Thus we see that the signal, S(n), is predictable using a linear combination of past outputs and the inputs. Hence the name linear prediction. This all-pole model formulation is equivalent to the autoregressive model in statistics.

The linear prediction model is shown in block diagram form in Fig. 2. The linear prediction residual (LPR) is defined as the sum of the squares of the error sequence e(n). Thus,

$$\text{LPR} \equiv \sum_{n} e^2(n) \qquad (2)$$



(a)



(b)

Fig. 2. (a) Discrete all-pole model in frequency domain.
(b) Discrete all-pole model in time domain.

For a given sequence of data $\{S_T(n)\}$ and using least squares estimation procedures, a set of linear prediction coefficients $a_T(k)$ can be calculated which minimize LPR. These coefficients form the well known least-squares estimator and satisfy the Töeplitz system of linear equations:

$$\begin{bmatrix} R_0 & R_1 & R_2 & \cdots & R_{11} \\ R_1 & R_0 & R_1 & \cdots & R_{10} \\ R_2 & R_1 & R_0 & \cdots & R_9 \\ \vdots & \vdots & \vdots & & \vdots \\ R_{11} & R_{10} & R_9 & \cdots & R_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{12} \end{bmatrix} = - \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_{12} \end{bmatrix} \tag{3}$$

A version of Levinson's method for inverting a Töeplitz matrix[27] is then used to solve for the twelve predictor coefficients, a_k, k = 1, 2, ... 12. Finally, since Eq. 3 was derived by minimizing, we derive the minimum prediction residual[24] (MPR) as:

$$\text{MPR} = \left[\sum_n e^2(n) \right]_{\min} = a_T' \, R_T \, a_T \tag{4}$$

where

a_T = augmented vector of predictor coefficients for a frame of the template; i.e., $a_T = \left[1 \; a_1 \; a_2 \; \cdots \; a_{12}\right]^T$

R_T = augmented matrix of autocorrelation coefficients for a frame of the template; i.e., $R_T = \left[R_{ij}(i - j)\right]$, i = 0, 1, ... 12, j = 0, 1, ... 12.

The prime indicates vector transpose.

This process is repeated for each frame of the template, and the value of $a_T' R_T a_T$ and the values of the elements of the vector R_T are stored for each frame. The number of frames, I, in the template

is calculated and this number stored also. These stored values provide all the information about the reference template that is required for comparison with the incoming speech. For a more extensive discussion of LPC analysis, see Makhoul.[26]

Incoming Speech Processing

The incoming speech is processed one frame at a time in the same way as the template to produce for each data frame a vector a_s of twelve predictor coefficients. Finally the product $a_s' R_T a_s$ is formed. Now, since Eq. 3 was arrived at by minimizing the prediction residual, we have defined a unique set of predictor coefficients, a_T, related to the given data sequence. Any other set of a's, say a_s (a_s is augmented as in Eq. 4), will yield a larger error residual. Thus we get

$$a_s' R_T a_s \geq a_T' R_T a_T \tag{5}$$

where the equality holds only if $a_s = a_T$. Thus,

$$\frac{a_s' R_T a_s}{a_T' R_T a_T} \geq 1 \tag{6}$$

Based on the above analysis, the ratio $a_s' R_T a_s / a_T' R_T a_T$ might be used as a distance measure to compare two speech waveforms. As the waveforms match better and better, the ratio becomes smaller and smaller, reaching a value of 1 for a perfect match.

Itakura has already demonstrated[22] that the ratio in Eq. 6 can be successfully used for comparing two speech waveforms in an isolated

word recognition experiment.

It is clear in Eq. 6 that the ratio is positive and from Eq. 5 that the ratio in Eq. 6 is greater than or equal to one. Therefore, if we define the frame similarity function, S, to be the inverse of the ratio in Eq. 6, we have a similarity function which satisfies the condition $0 \leq S \leq 1$ and thus allows us to use Bridle's time-warp algorithm, without modification. While both Itakura[22] and Coker[25] use the log of the ratio in Eq. 6, the log was not used in this system since the ratio could be used directly in Bridle's algorithm.

$$S = \frac{a_T' R_T a_T}{a_s' R_T a_s} \tag{7}$$

In summary, we have a ratio whose numerator, $a_T' R_T a_T$, is the minimum prediction residual for a frame from the reference template. The denominator, $a_s' R_T a_s$, is of the same form as the numerator but involves the predictor coefficients from a frame of input speech and the autocorrelation coefficients from a frame of the template.

Thus the denominator is always greater than the numerator except when $a_s = a_T$ for which the numerator and denominator are equal. If the two frames are from similar speech, then we expect S to be close to one, and if they are from dissimilar speech, we expect the ratio to be closer to zero. Thus, S should be a reasonable measure of similarity between different segments of speech.

Time-Warp Algorithm

To explain the time-warp algorithm, first let us consider the notion of a time-registration path as shown in Fig. 3.

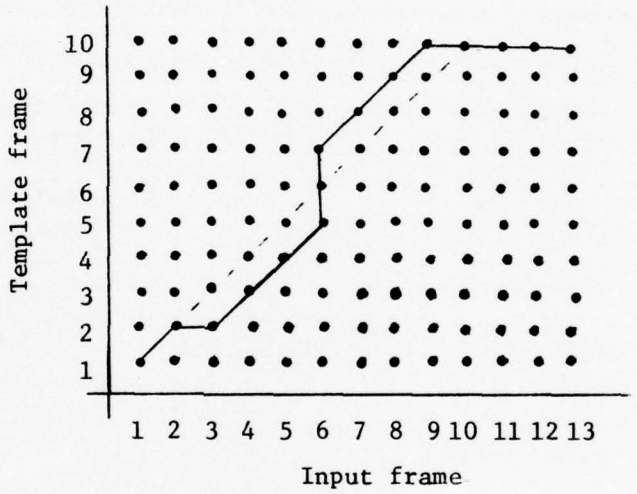


Fig. 3. Time registration diagram.

For this discussion let us assume that we have a template word that is ten frames long, and it is being time-registered to a version of the word that is thirteen frames long. Figure 3 illustrates a possible time-registration path. Each frame of the template has a row of dots, each dot corresponding to a possible match between that template frame and a frame of input speech. Similarly, each frame of input speech has a column of dots with each dot corresponding to a possible match between the input frame and each frame of the template.

A possible time-registration path is represented by the solid

line connecting neighboring dots in a horizontal, vertical, or diagonal direction. In this diagram, then, it is seen that frame 2 of the template is being compared with both frames 2 and 3 of the input speech. Frame 6 of the input speech is being compared with frames 5, 6, and 7 of the template, and frame 10 of the template is being compared with frames 9 through 13 of the input speech. If time-warping were not done, we would only compare the ten template frames with ten consecutive input frames. This comparison is represented by the broken diagonal line in Fig. 3.

The time-warp algorithm uses a dynamic programming approach to perform a recursive search for a possible time-registration path along which the two waveforms are similar. This procedure has the effect of stretching or compressing the input waveform at various points. The criteria which define an acceptable path ensure that the template and the incoming speech are similar throughout their lengths. This similarity is adjusted by varying G, Q, and K as described below.

To describe Bridle's time-warp algorithm,[23] let us first define a local similarity function for any point (i, j) in a time-registration diagram by:

$$L(i, j) = (1 - G)\, S(i - a, j - b) + KGS(i, j)$$

where

i corresponds to the template

j corresponds to the input speech

(i - a, j - b) corresponds to a possible previous point

$$(a, b) = \begin{cases} (0, 1) \text{ horizontal} \\ (1, 0) \text{ vertical} \\ \text{or} \\ (1, 1) \text{ diagonal} \end{cases}$$

G is an exponential weighting "time constant" which forces the local similarity, L(i, j), to depend more on a match between the input speech and the template in the recent "past" than in the distant "past".

K is a time distortion penalty factor defined by:

$$K = \begin{cases} 1 & \text{if } (a, b) = (1, 1); \text{ i.e., no time warp} \\ k & 0 \leq K \leq 1, \text{ if } (a, b) = (1, 0) \text{ or } (0, 1) \end{cases}$$

The parameter K is introduced to penalize any time distortion but is utilized so as to allow poorly matching frames if they occur between other frames which match well. Since there are no previous points for the first point on the path, we define $L(1, j) = S(1, j)$.

We then decide that the template word is present in the input speech if the local similarity function is greater than some threshold, Q, at every point along the path.

Here it should be noted that L(i, j) is an exponentially weighted sum of values of L from previous points where the threshold was exceeded. Thus, the effect of an isolated pair of dissimilar frames can be overcome by nearby pairs of frames which are similar. This allows the system to find words whose waveforms are substantially similar but have some short segments that are dissimilar.

The local similarity function, L(i, j), is then used in a recursive search algorithm which searches for all acceptable time-registration paths which could match the template to the incoming speech. To accomplish this, let us define a detection function A(i, j). If there is no path to the point (i, j) for which L exceeds the threshold, Q, at every point along the path, then set A(i, j) = 0. Otherwise set A(i, j) = maximum L(i, j), where L(i, j) is maximized over all paths to (i, j) for which L > Q at every point. For a given input speech frame, j, a value of A(i, j) is computed for each frame, i, of the template. After an input frame has been compared to each frame of the template of length I frames, a test is made on A(I, j). If A(I, j) > 0, then the system indicates that the template word was present in the incoming speech ending at frame j.

This procedure does not find the best matching path out of all possible paths but merely determines whether there are any acceptable paths. As a result it may find that A(I, j) > 0 for several successive values of j.

The search algorithm is implemented computationally in the following way.

Let us define a new function

$$\text{Step } (A, S, K) = \begin{cases} 0, \text{ if } A = 0 \\ 0, \text{ if } (1 - G)A + KGS < Q \\ (1 - G)A + KGS, \text{ otherwise} \end{cases}$$

where A, S, and K were defined previously. Then set

$$A(i, j) = \max_{(a,b)} \{\text{Step } [A(i - a, j - b), S(i, j), K(a, b)]\}$$

where (a, b) takes the values (1, 0), (0, 1), and (1, 1); i > 1; and the corresponding values of K(a, b) are k, k, 1.

If i = 1,

$$A(1, j) = \max \{\tilde{S}(1, j), \text{step } [A(1, j - 1), S(1, j), K]\}$$

where

$$\tilde{S}(1, j) = \begin{cases} 0, & \text{if } S(1, j) < Q \\ S(1, j), & \text{if } S(1, j) > Q \end{cases}$$

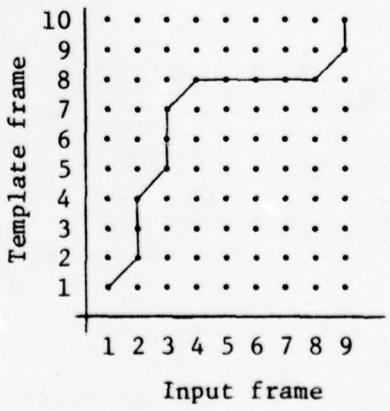
Thus, we see that although A(i, j) is defined in terms of all possible paths to (i, j), it can also be defined in terms of A(i - 1, j), A(i - 1, j - 1), A(i, j - 1), and S(i, j). We then compute three quantities which correspond to arriving at the point (i, j) from the left, from below, and from diagonally below (see Fig. 3), and then choose the maximum of the three values. This value is then stored in the A array at location A(i, j).

In order to illustrate what happens using the time-warp algorithm, some data from a real experiment using a threshold of 0.275 is reproduced in Fig. 4. Figure 4a shows the table of values which propagate through the array. Figure 4b shows the corresponding time registration path from a template of length ten frames being matched to a word of length nine frames.

In summary, we note the following:

Input Frame Number	Value at Each Location in Array A									
	1	2	3	4	5	6	7	8	9	10
1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.343	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.394	0.389	0.396	0.304	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.471	0.637	0.496	0.391	0.318	0.000	0.000	0.000
6	0.000	0.000	0.304	0.661	0.521	0.452	0.572	0.321	0.000	0.000
7	0.000	0.000	0.000	0.617	0.738	0.614	0.718	0.371	0.000	0.000
8	0.000	0.000	0.000	0.391	0.733	0.787	0.562	0.388	0.000	0.000
9	0.000	0.000	0.000	0.379	0.620	0.767	0.811	0.370	0.000	0.000
10	0.000	0.000	0.000	0.304	0.334	0.420	0.593	0.406	0.000	0.000
11	0.328	0.000	0.000	0.000	0.000	0.000	0.300	0.523	0.521	0.376

(a)



(b)

Fig. 4. (a) Table of values in A array; (b) time registration path corresponding to Fig. 4a.

1. The input speech is processed one frame at a time.
2. All necessary information about previous frames is contained in the A array.
3. Each j such that A(I, j) > 0 is identified as the last frame of the word to be spotted.

Multiple Templates

All of the above discussion assumes the use of only one template. However, if one could somehow make use of several templates combined in some appropriate way into a composite template, improved accuracy and flexibility might be achieved. One might use several templates separately with a decision based on a majority vote. Another alternative is a nearest neighbor comparison on a frame-by-frame basis. The second approach was chosen for this work. Motivation for this choice is provided in the following discussion.

In standard pattern recognition experiments, we have at least three classification cases to deal with. A brief description of each case follows.

Case 1: We know the a priori probabilities $P(\omega_j)$ and the class conditional densities $p(x/\omega_j)$, where x is the observed feature and ω_j is the true state of nature. In this case we can design an optimal classifier using Bayes' decision theory.

Case 2: We only know the forms of the density functions. In this case we use the samples to perform maximum

likelihood or Bayesian estimates of the true parameter values.

Case 3: The only information available is a set of samples. We do not know anything about the forms of the underlying densities. In this case we use nonparametric techniques to estimate the densities. Or we can use the nonparametric nearest neighbor decision rule to bypass the probability estimation and go directly to a decision function. If we have an unlimited number of samples, it can be shown that the error rate is never worse than twice the Bayes rate.[34] The nearest neighbor procedure is defined as follows:

> Let $H^n = \{x_1, x_2, \ldots x_n\}$ be a set of labelled samples and let $x'_n \in H^n$ be the sample nearest to x. Then the nearest neighbor rule for classifying x is to assign it the label associated with x'_n.

In the word spotting problem, we have a Case 3 pattern-recognition problem, only instead we have only a few labelled samples from one class or word, and we are interested in finding whether the observed sample of incoming speech belongs to that class. To accomplish this we require that the observed speech be closer to the template word than some threshold using the algorithm described above in the time-warp algorithm section.

To see how the multiple templates are used, suppose there are

M templates. The initial processing on each template is done as before. Then a frame similarity function $S_m(i, j)$ is calculated by comparing the ith frame of the mth template with the jth input speech frame. If i exceeds the number of frames in the mth template, then $S_m(i, j)$ is set to zero. Now let

$$S(i, j) = \max_{m = 1, 2, \ldots M} S_m(i, j)$$

Then use S in the remainder of the algorithm as before. This approach has the effect of dynamically creating a composite template each time it begins to match up with the incoming speech. The effect of this procedure along with the automatic gain control effect of normalizing the autocorrelation coefficients and the time warp procedure is to make two or three templates look like a large sample. For example, suppose we had three templates each three frames long. Using the above procedure there are 27 possible ways the incoming speech might match up. Thus this multiple template procedure should be superior to the majority vote procedure. A brief preliminary experiment using three templates confirmed this conclusion.

To illustrate this nearest neighbor procedure, let us assume we have three templates with corresponding lengths of three, four, and five frames. The three templates are shown in block form in Fig. 5 with an x marked in the frame of the template which yielded the maximum value for $S_m(i, j)$. The equivalent or effective composite template is shown at the bottom where the template number which was

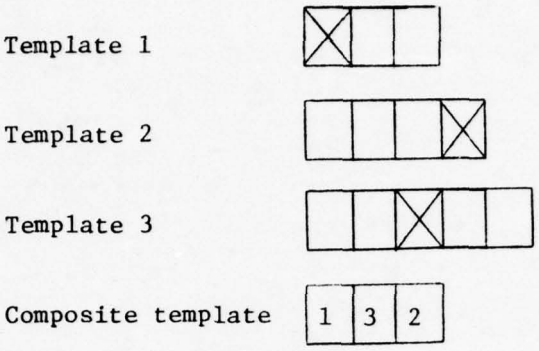


Fig. 5. Composite template construction.

used to create each frame of the composite is indicated.

V. EXPERIMENTAL RESULTS

Introduction

In the following section the experimental results are summarized. It should be noted that in all cases two parameters were adjusted or tuned to yield minimum error performance. These were threshold, Q, and time distortion penalty, RK. In the initial experiments, it was found empirically that setting the exponential factor, G, to a value of 0.3 yielded the best results in most cases, and so it was set at that value for this work.

For these experiments three different types of text were used. These types of text will be labeled as PDP-11 (Appendix II), Ford (Appendix III), and Digits, from a table of random numbers. A total of seven basic experiments were conducted as described below, using a total of 1651 key words of which 1217 were the digits zero through nine. A key word is a word to be spotted which is actually present in the text. For example, in the PDP-11 text, the word "interrupt" occurs 18 times, "priority" occurs 6 times, "program" occurs 10 times, and "processor" occurs 7 times. In the experiments we were trying to spot those four words so there are 41 key words in the PDP-11 passage. If we use the PDP-11 passage spoken by two people, then there are 82 key words.

In word spotting there are two types of errors, false alarms

and misses. For false alarms the quantity of primary interest is the false alarm rate per unit time. Accuracy is then defined as the percentage of occurrences of the word actually found. That is,

$$C_1 = \frac{NS}{T} \times 100$$

where

C_1 = word spotting accuracy

T = total number of occurrences of the word

NS = number of times the correct word was spotted

However, the digit recognition experiments have a different objective than the standard word spotting. In this case false alarms and misses are equally bad errors and are therefore given the same weight. Therefore we define accuracy in digit recognition as

$$C_2 = \frac{NS - F}{T}$$

where

C_2 = digit recognition accuracy

T = total number of occurrences of the digit being recognized

NS = number of times the correct digit was recognized

F = number of false alarms

In addition to the seven basic experiments, some additional experiments of word spotting in noisy long distance radio speech with both telephone and radio channel distortion were performed. Also, a brief experiment was performed in which a template word was constructed

from parts of other words spoken by the same speaker. These last experiments were performed merely to get an indication of performance and will be reported only from that point of view and to suggest further work.

In the description of experiments presented below, each experiment is given a number and is described separately.

Experiments

Experiment 1

In order to be able to compare results using this word spotting procedure with those reported by Bridle and Brown, it was decided to use the same text and words in the initial experiments as were used by Bridle. This is the PDP-11 passage (see Appendix II). This passage takes about two minutes to read. The passage was read by five speakers, four male and one female. In this experiment the template words were then spoken in isolation by each speaker in contrast to Bridle's procedure where he extracted the templates from the speech text. By having the template words spoken in isolation rather than taken from the text, coarticulation effects are not present in the templates as they were in Bridle's case. Coarticulation effects can occur, for example, when the vocal tract is changed or modified during the pronunciation of a word to allow a smooth transition to occur from one word to the next. Thus coarticulation effects vary with context, speed of talking, location of the word in a sentence, emphasis, etc.

Since coarticulation effects are subject to so many variables,

it was decided to have the template words spoken in isolation but still have them spoken several times with the natural variability of talking speed.

Thus, the speakers read the text and spoke the words as they naturally would without coaching, except to say the template words in isolation.

The words to be spotted were "interrupt", "program", "processor", and "priority". Word spotting accuracy was greater than 99 percent with no false alarms. The combined occurrence of all the key words was 205 in the five passages. In this experiment three templates spoken by the same speaker as the text were used for spotting each word. Actual results are shown in Table 1.

Experiment 2

This experiment was a cross-speaker word-spotting experiment using the PDP-11 text. In this case six templates were used for each word, three from speaker No. 1, three from speaker No. 2, and the text was from two other speakers. In this case the same four words as in Experiment 1 were spotted with an accuracy of 94 percent and no false alarms. In this case the combined occurrence of the key words was 82.

In order to illustrate the trade-off between accuracy and false alarm rate, this experiment was also done so as to yield 100 percent accuracy. In this case there were 11 false alarms, yielding a false alarm rate of approximately 180 per hour. The results of this experiment are shown in Table 2.

Table 1. Word spotting results from Experiment 1.

Text Speaker No.	Word	Threshold Q	Time Wave Penalty Factor RK	Misses	False Alarms Per Hour
1	Interrupt	.54	.55	0	0
	Processor	.47	.55	0	0
	Program	.47	.55	0	0
	Priority	.47	.55	0	0
2	Interrupt	.43	.70	0	0
	Processor	.43	.70	0	0
	Program	.43	.70	0	0
	Priority	.43	.70	0	0
3	Interrupt	.50	.63	0	0
	Processor	.54	.63	0	0
	Program	.50	.63	0	0
	Priority	.51	.63	0	0
4	Interrupt	.45	.60	0	0
	Processor	.45	.60	0	0
	Program	.46	.60	0	0
	Priority	.45	.60	0	0
5	Interrupt	.47	.65	0	0
	Processor	.47	.65	0	0
	Program	.47	.65	2	0
	Priority	.47	.65	0	0

Table 2a. Cross-speaker word spotting results from Experiment 2 (adjusted for zero false alarm rate).

Template Speaker No.	Word	Text Speaker No.	Threshold Q	Time Warp Penalty Factor RK	Misses	False Alarms Per Hour
1 and 2	Processor	5	.46	.7	0	0
1 and 2	Processor	6	.50	.7	0	0
1 and 2	Priority	5	.53	.7	1	0
1 and 2	Priority	6	.45	.4	1	0
1 and 2	Program	5	.48	.6	0	0
1 and 2	Program	6	.46	.3	1	0
1 and 2	Interrupt	5	.55	.4	1	0
1 and 2	Interrupt	6	.50	.6	1	0

Table 2b. Cross-speaker word spotting results from Experiment 2 (adjusted for 100 percent accuracy).

Template Speaker No.	Word	Text Speaker No.	Threshold Q	Time Warp Penalty Factor RK	Misses	False Alarms Per Hour
1 and 2	Processor	5	.46	.7	0	0
1 and 2	Processor	6	.50	.7	0	0
1 and 2	Priority	5	.50	.7	0	1
1 and 2	Priority	6	.43	.2	0	1
1 and 2	Program	5	.48	.6	0	0
1 and 2	Program	6	.45	.3	0	2
1 and 2	Interrupt	5	.52	.6	0	2
1 and 2	Interrupt	6	.45	.3	0	5

Experiment 3

In this experiment the PDP-11 text spoken by speaker No. 1 was used. Here the objective was to evaluate the system performance in the presence of additive white gaussian noise and to try three different techniques to perform the word spotting using the noisy speech.

The noisy speech was generated as shown in the block diagram, Fig. 6.

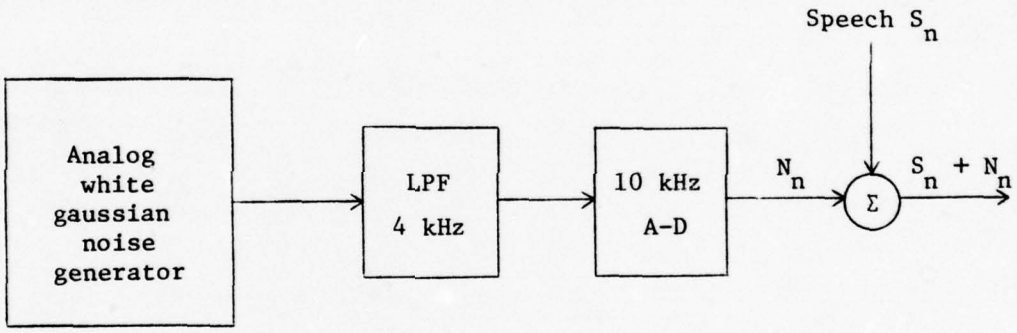
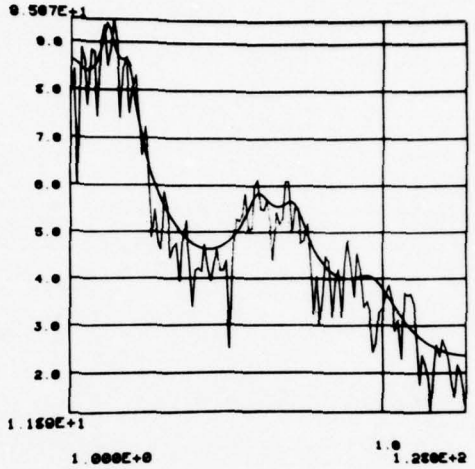


Fig. 6. Noisy speech generator.

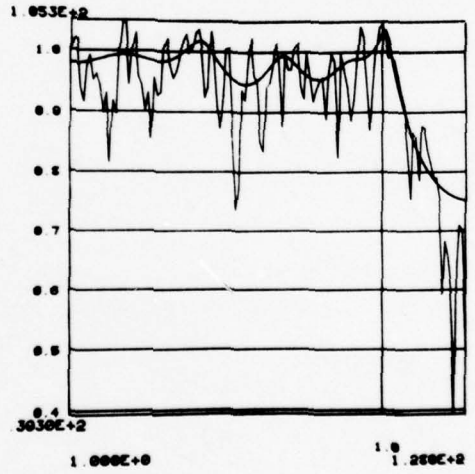
The spectrum of 12.8 ms of noise is shown in Fig. 7a, where the heavy curve is the LPC smoothed spectrum and the light curve is the FFT spectrum. A similar spectrum for 12.8 ms of speech from the word "Ford" is shown in Fig. 7b.

The signal-to-noise ratio, SNR, was calculated as follows:

$$\text{SNR} = 10 \log_{10}\left(\frac{\sum_{n=1}^{256} S_n^2}{\sum_{n=1}^{256} N_n^2}\right) \qquad (1)$$



(a)



(b)

Fig. 7. (a) Speech spectrum; (b) noise spectrum.

Thus a value for SNR for each block of 256 samples was defined. Then a measure of average SNR for the whole passage of speech was defined as:

$$\overline{\text{SNR}} = 10 \log_{10} \sum_{k=1}^{K} \left[\frac{\left(\sum_{n=1}^{256} S_n^2 \right)_k}{\left(\sum_{n=1}^{256} N_n^2 \right)_k} \right] \qquad (2)$$

where

n = sample number within a block of 256 samples

k = block number

K = number of blocks in the whole passage

Here it should be noted that this SNR is an average value over the frequency spectrum of the speech. Actually the SNR varies as a function of frequency over the frequency spectrum of the speech signal. In particular, the SNR is higher at the lower frequencies where most of the speech energy is present.

Since the measure of similarity used for the word spotting is calculated using a 256-sample block of data and is also an average measure over the frequency spectrum of the speech, it was concluded that this particular definition of SNR was appropriate.

It should also be noted that the main objective in the noisy speech experiments was to determine whether the system degraded smoothly or abruptly in the presence of noise, and at what average SNR the performance changed. Three procedures for doing the word spotting in the

presence of noise were to be tried. It was not the objective to gain an in-depth understanding of all the pertinent noise issues and to perform an in-depth evaluation of each procedure. The in-depth evaluation and understanding of the noise issues should be the product of a separate research project.

The objective here was to point some directions and suggest some things to be done to pursue the noise questions. Performance degradation in the presence of noise is shown in Fig. 8, where it is evident that performance degrades smoothly and does not occur until $\overline{SNR}$ is less than 24 dB. In Fig. 8, the accuracy curve is drawn for a false alarm rate less than or equal to 66 false alarms per hour to allow for meaningful comparisons.

Autocorrelation Method

In this method we calculate the autocorrelation of the speech plus noise, R_{S+N}, and the autocorrelation of the noise R_N and then take the difference to get $\hat{R}_S = R_{S+N} - R_N$ where

$$R_{S+N}\Big|_i = \sum_{n=0}^{N-1-|i|} (S_n + N_n)(S_{n+|i|} + N_{n+|i|}) \tag{3}$$

$$R_N\Big|_i = \sum_{n=0}^{N-1-|i|} N_n N_{n+|i|} \tag{4}$$

Now, if the speech and the noise signals are uncorrelated, the expected value of the cross terms in Eq. 4 should be zero. Thus $\hat{R}_S = R_{S+N} - R_N$

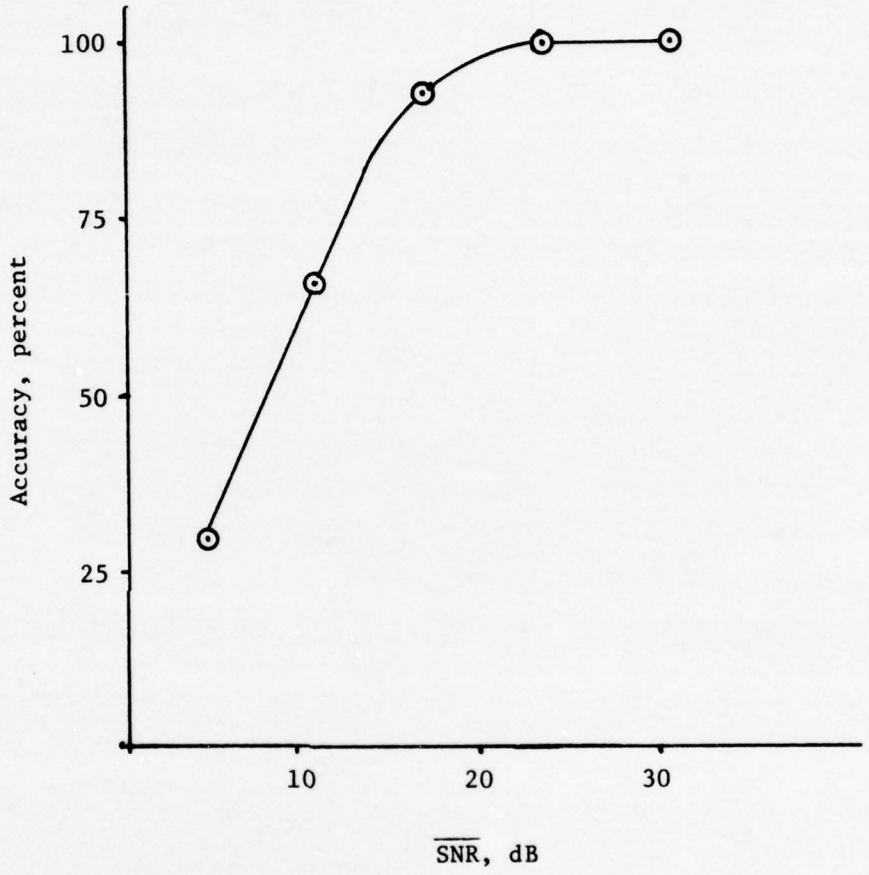


Fig. 8. Performance in presence of noise.

might yield a good estimate of R_S. One major problem with this approach is that the Töeplitz matrix generated from $\hat{R}_S$ might not be positive definite, whereas the matrix inversion procedure requires the matrix to be positive definite. This could occur because only 256 samples are being used to generate the autocorrelations, and this may not be sufficiently long to yield zero values for the cross terms in Eq. 3 above.

This approach was tried with speech having $\overline{SNR}$ = 11 dB and yielded an accuracy of 51 percent compared with 66 percent before processing (see Table 3). When speech with $\overline{SNR}$ = 4.6 dB was used, the similarity measure was greater than 1, indicating clearly that the calculation of the linear prediction residual was incorrect.

Wiener Filter Method[29]

In this procedure, an average spectrum of the noise is calculated using many blocks of noise with the same statistics as the noise added to the speech. This spectrum is then used to construct a new Wiener filter for each block of 256 samples of speech plus noise. The output of the Wiener filter is an estimate of the true speech signal. For a block diagram description, see Fig. 9.

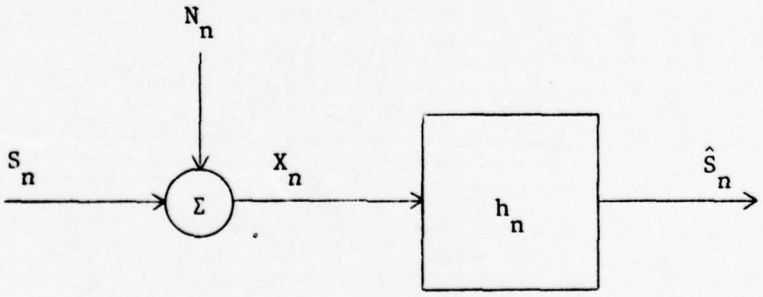


Fig. 9. Wiener filter block diagram.

In Fig. 9, $h_n = F^{-1}\{H(\omega)\}$ where $H(\omega) = \left(\Phi_{xx}(\omega) - \Phi_{nn}(\omega)\right)/\Phi_{xx}(\omega)$. For this experiment a previously developed Wiener filter computer program was used to process the noisy speech.

This Wiener filtering approach was tried on the noisy speech

with $\overline{SNR}$ = 11 dB and $\overline{SNR}$ = 4.6 dB. For $\overline{SNR}$ = 11 dB, an accuracy of 95 percent was achieved compared with 66 percent without Wiener filtering. For $\overline{SNR}$ = 4.6 dB, an accuracy of 68 percent was achieved compared with 29 percent without the filtering. These results are summarized in Table 3 for comparison with other methods.

Table 3. Noisy speech word spotting results. (The false alarm rate for all cases is 66/hour.)

$\overline{SNR}$	No Processing to Remove Noise Effects	Subtract Autocorrelation of Noise	Design Wiener Filter	Noisy Template
11	66	51	95	90
4.6	29	*	68	68

* These results were not meaningful.

Since $\Phi_{xx}(\omega) = F\{R_{xx}\}$ and $\Phi_{nn}(\omega) = F\{R_{nn}\}$, one might ask why the Wiener filter approach works so much better than the simpler and faster autocorrelation approach. On the surface, the two procedures appear to do about the same thing except one is in the time domain and the other is in the frequency domain.

In the autocorrelation experiments, nothing was done to force the autocorrelation matrix to be positive definite. In the Wiener filter algorithm, $H(\omega)$ was never allowed to become negative. Instead, as $H(\omega)$ approached zero, its amplitude was changed using a smooth tailoring curve so that it approached zero but could not become

negative.

Noisy Template Method

In this method, noise with the same statistics as that in the noisy speech was added to the template. Motivation for this procedure is provided in the following discussion.

If the noise spectrum were perfectly flat, as would be the case with white gaussian noise, then the speech spectrum in the passage and templates should just be biased upward by a constant factor at all frequencies. In this system, what we are really doing is comparing the short time spectrum of the incoming speech with that of a template. Suppose that these two spectra look alike in the absence of noise. If we just add the same flat spectrum noise to both the template and incoming speech, then we would expect the two spectral envelopes to still look alike, only biased upward in amplitude. However, there is no such thing as real white gaussian noise, and the noise spectrum of our real noise is not flat over even relatively short frequency spans as can be seen in Fig. 7. Even so, the noise spectrum is flat enough that we do expect improvement if noise is added to the word templates.

For the passage with $\overline{SNR}$ = 11 dB, the accuracy was 90 percent compared to 66 percent without additional processing. For the passage with $\overline{SNR}$ = 4.6 dB, the accuracy was 68 percent compared to 29 percent without additional processing. These noise results are summarized in Table 3 for comparison purposes.

Experiment 4

Words used up to this point have been multiple syllable words. One might question how well the system would work on very short one syllable words. To address this question and at least get an indication of performance with short words, a passage of text containing the word "Ford" was constructed from a news magazine (see Appendix III). The word "Ford" occurred twelve times and the passage took approximately one minute to read. As before, the template words were read in isolation. Two templates were used.

Performance accuracy was 100 percent with no false alarms. When the threshold was lowered until false alarms occurred, the first false alarms occurred on sounds like "org" in "organization" and "for D" as in "for Donald" as one might expect.

A very limited experiment was performed with a short segment of this passage (approximately 10 s) to get an indication of performance degradation in the presence of noise for short words. These preliminary results indicate that performance begins to degrade at a signal-to-noise ratio which is 6 to 10 dB higher than in Experiment 3 above.

Experiment 5

To test the cross-speaker performance on short words, the Ford text was used with templates provided by different individuals. This experiment was performed using six templates from six different individuals, then five, then four, then three, then two, where the two

were a subset of the three, the three a subset of the four, etc.

The resultant accuracy was 100 percent with no false alarms in all cases, as shown in Table 4. Here it should be noted that as the

Table 4. Cross-speaker word spotting with Ford text.

Number of Templates	Q	RK	Accuracy Percent
6	.55	.55	100
5	.55	.55	100
4	.50	.55	100
3	.45	.55	100
2	.44	.55	100

number of templates was increased, the threshold also had to be increased.

Experiment 6

Finally, in order to make some comparisons with previous work in recognizing digits, a digit-recognition experiment was performed. In this experiment, ten speakers -- five males and five females -- each spoke approximately one hundred digits chosen from a table of random numbers. The digits used were zero through nine. Speaker No. 2 spoke two sets of digits. One set was deliberately spoken in isolation and the second was spoken with random speeds, emphasis, and groupings of numbers. The remaining speakers read the digits from the

table as they would normally read such a list over the telephone. The results of this experiment are presented in Table 5.

Table 5. Digit recognition results.

Speaker	Number of Digits	False Alarms	Misses	Percent Accuracy
1	100	0	0	100
2	98	0	4	96
2	119	0	2	98
3	96	1	0	99
4	100	0	0	100
5	100	0	2	98
6	100	1	1	98
7	100	6	14	80
8	100	0	0	100
9	100	0	2	98
10	106	2	2	96

As can be seen, the accuracy varied from 80 percent to 100 percent on an individual basis, with an overall accuracy of 97 percent. However, only speaker No.7 departs significantly from 100 percent, with the next lowest accuracy being 96 percent.

The factor which appeared to affect accuracy the most was articulation, that is, how clearly and distinctly each digit was articulated. For example, if the "t" sound in "eight" was not present,

then the "eight" digit was confused with "three", etc. Speaker No. 7 yielded the poorest results but also had the greatest apparent variability in speaking. For example, sometimes she rolled the "r" in "three" and sometimes she did not. The word "eight" sometimes had the "t" sound and sometimes did not.

In general, it was found that when the digits were spoken clearly and distinctly, the system performed near 100 percent accuracy.

Experiment 7

This experiment was a very brief attempt to do cross-speaker digit recognition to give an indication of performance. In this case two templates from speaker No. 4 were used on the digits from speaker No. 1. The resulting accuracy was 82 percent, with the errors divided almost equally between false alarms and misses. Speakers No. 1 and 4 were chosen because they spoke the digits most alike. Again, the failures tended to occur when words were not articulated clearly.

In addition to the experiments described above, some additional tests of the system were performed which give some indication of the robustness of the system. In one experiment a recording was made on an inexpensive cassette recorder of some noisy speech of rather poor quality from a telephone talk show broadcast from a distant AM radio station. This recording included the distortion and noise introduced by the telephone, the radio, and night-time ionospheric effects.

The radio speech was then processed to remove convolutional distortion using a blind deconvolution procedure developed by Stockham.[38] Using multiple templates constructed from the incoming speech, an accuracy of 100 percent was achieved on two separate passages with two different speakers saying the words in each passage. The words used in these two cases were "dream" and "grand".

Finally, inasmuch as one might not be able to get a template word from a given speaker with unique speech characteristics, an experiment was performed in which a template was constructed using parts of other words spoken by the same speaker. In this case, the speech was also poor quality radio speech from another distant late night talk show. The word to be constructed was "restaurant". The experiment was done rather crudely just to get an indication of results. The word "restaurant" was constructed using the "r" sound from "respect", the "est" sound from "best", and the "ant" sound from "want".

Using only one template so constructed, an accuracy of 80 percent was achieved. Since this effort was not part of the main objective of the research, it was not pursued any further.

VI. CONCLUSION

A word spotting system has been demonstrated that works remarkably well using only a single measure of similarity between the incoming speech and the template word. It uses a simple procedure to dynamically construct a composite template from a set of multiple templates. The "best" performance of the system, using only this single measure of similarity, has been demonstrated by adjusting two parameters to tune the system for each word and speaker.

The results clearly indicate that it is reasonable and feasible to use LPC parameters in word spotting. The time-warp algorithm of Bridle was very effective and efficient and allowed correct word spotting when the word and template differed in length by as much as a factor of two.

Once the system was implemented and working, the majority of nonprocessing time was taken up in isolating the templates. In an operational system, one would certainly want to have a reliable fully automatic endpoint detector to isolate the templates.

The results using noisy speech were very encouraging. In particular, the success achieved using the poor quality distorted radio recordings indicates that this approach has great promise for future use in that environment.

Although this system as it now exists would probably not be acceptable for operational use in either word spotting or digit recognition, it does point the direction for further research to develop such a system.

The main disadvantage of the system as it now exists is the need to readjust the threshold and time-distortion penalty factor for each word and speaker.

Some of the main advantages of this approach are the simplicity and speed of the analysis and that one could use a real-time LPC vocoder in generating the similarity measure. Another advantage is that additional features could easily be added to the similarity measure in hopes of improving the cross-speaker performance and decreasing the strong dependence of performance on the similarity threshold. The system also performs well when the speech is contaminated by additive white gaussian noise.

Areas for Future Research

There are many areas of additional research needed to improve the performance and robustness of this system. Some of these are listed below:

1. Add features such as voicing, pitch contour, etc., to the similarity measure to reduce the need to readjust the threshold, improve noisy speech performance, and improve cross-speaker performance.
2. Experiment with possible techniques to calculate the

threshold for each word and/or speaker. For example, one might use some of the short time speech statistics.[28]

3. Experiment with additional front-end processing such as preemphasis to boost the high frequency portion of the spectrum.
4. Conduct research to improve performance by changing the window shape and/or length, varying the sampling rate, varying the number of poles in the LPC analysis, adding zeros in the LPC analysis, etc.
5. Address the noise and distortion problem using other kinds of noise and distortion and other procedures such as adaptive noise canceling.[30] In addition, conduct more detailed and definitive research into the applicability and use of deconvolution procedures such as the blind deconvolution procedure of Stockham.[38]
6. Conduct research to normalize the frequency domain to improve the cross-speaker performance. For example, one might normalize the vocal tract area or length and thus be able to use the same templates for both male and female speakers. Wakita[35-37] has reported encouraging results using vocal tract length normalization. His hypothesis was that speech sounds produced by arbitrarily selected speakers which are perceptually categorized into an identical phoneme result from similar vocal tract shapes of different lengths. His preliminary experiments yielded much more

compact distributions of the first three formants than are achieved without the normalizations. The possibility of some equivalent normalization in the time domain, possibly using reflection coefficients, should be examined.

REFERENCES

1. K. H. Davis, et al., "Automatic Recognition of Spoken Digits", *Journal of the Acoustical Society of America*, Vol. 24, No. 6, November 1952.

2. J. Wiren and H. L. Stubbs, "Electronic Binary Selection System for Phoneme Classification", *Journal of the Acoustical Society of America*, Vol. 23, No. 6, November 1956.

3. H. F. Olson and H. Belar, "Phonetic Typewriter", *Journal of the Acoustical Society of America*, Vol. 28, No. 6, November 1956.

4. H. F. Olson and H. Belar, "Phonetic Typewriter III", *Journal of the Acoustical Society of America*, Vol. 33, No. 11, November 1961.

5. H. Dudley and S. Balashek, "Automatic Recognition of Phonetic Patterns in Speech", *Journal of the Acoustical Society of America*, Vol. 30, No. 8, August 1958.

6. H. Dudley, "Phonetic Pattern Recognition Vocoder for Narrowband Speech Transmission", *Journal of the Acoustical Society of America*, Vol. 30, No. 8, August 1958.

7. J. W. Forgie and C. D. Forgie, "Results Obtained from a Vowel Recognition Computer Program", *Journal of the Acoustical Society of America*, Vol. 31, No. 11, November 1959.

8. P. Denes and M. V. Mathews, "Spoken Digit Recognition Using Time-Frequency Pattern Matching", *Journal of the Acoustical Society of America*, Vol. 32, No. 11, November 1960.

9. D. R. Reddy, "Computer Recognition of Connected Speech", *Journal of the Acoustical Society of America*, Vol. 42, No. 2, 1967.

10. B. Gold, "Computer Program for Pitch Extraction", *Journal of the Acoustical Society of America*, Vol. 34, 1962.

11. R. F. Purton, "Speech Recognition Using Autocorrelation Analysis", *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-16, No. 2, June 1968.

12. J. N. Shearme and P. F. Leach, "Some Experiments with a Simple Word Recognition System", *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-16, No. 2, June 1968.

13. P. Ladefoged, I. Kameny, and W. A. Brackenridge, "Acoustic Effects of Style of Speech", *88th Meeting of Acoustical Society of America*, November 1974.

14. R. A. Gillman, "Automatic Verification of Hypothesized Phonemic Strings in Continuous Speech", TM-5315/000/00, S.D.C., May 10, 1974.

15. R. N. Weeks, "Predictive Syllable Mapping in a Continuous Speech Understanding System", *IEEE Symposium on Speech Recognition*, Carnegie-Mellon University, April 1974.

16. R. A. Gillman, "Automatic Recognition of Nasal Phonemes", *IEEE Symposium on Speech Recognition*, Carnegie-Mellon University, April 1974.

17. L. Molho, "Automatic Recognition of Fricatives and Plosives in Continuous Speech", *IEEE Symposium on Speech Recognition*, Carnegie-Mellon University, April 1974.

18. R. Schwartz and J. Makhoul, "Where the Phonemes Are: Dealing with the Ambiguity in Acoustic-Phonetic Recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, February 1975.

19. L. J. Gerstman, "Classification of Self-Normalized Vowels", *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-16, No. 1, March 1968.

20. J. Makhoul, "Selective Linear Predictive Spectral Matching", *86th Meeting of the Acoustical Society of America*, Los Angeles, California, October 1973.

21. C. F. Teacher, H. G. Kellett, and L. R. Focht, "Experimental, Limited Vocabular, Speech Recognizer", *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-15, No. 3, September 1967.

22. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, No. 1, February 1975.

23. J. S. Bridle and M. D. Brown, "An Experimental Automatic Word-Recognition System", Interim Report, Joint Speech Research Unit, Middlesex, England.

24. S. F. Boll, "Waveform Comparison Using the Linear Prediction Residual", Computer Science Technical Memorandum No. 7500, University of Utah, May 1975.

25. M. J. Coker, "An Isolated-Word Recognition System Based on Linear Prediction Analysis", University of Utah Master of Science thesis, December 1975.

26. J. Makhoul, "Linear Prediction: A Tutorial Review", *Proceedings of the IEEE*, Vol. 63, No. 4, April 1975.

27. N. Levinson, "The Wiener RMS Error Criterion in Filter Design and Prediction", *Journal of Mathematics and Physics*, Vol. 25, No. 4, 1947, pp. 261-278.

28. L. R. Rabiner and M. R. Sambur, "Some Preliminary Experiments in the Recognition of Connected Digits", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-24, April 1976.

29. N. Wiener, *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*, John Wiley and Sons, Inc., New York, 1949.

30. B. Widrow, et al., "Adaptive Noise Cancelling: Principles and Applications", *Proceedings of the IEEE*, Vol. 63, No. 12, December 1975.

31. S. F. Boll, "A Priori Digital Speech Analysis", Ph.D. dissertation, University of Utah, March 1973.

32. J. R. Pierce, "Whither Speech Recognition?", *Journal of Acoustical Society of America*, Vol. 46, No. 4 (Part 2), April 1969.

33. G. Fant, *Speech Sounds and Features*, The MIT Press, Cambridge, Massachusetts, and London, England, 1973.

34. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York, 1973.

35. H. Wakita, "An Approach to Vowel Normalization", 89th Meeting of the Acoustical Society of America, Austin, Texas, April 1975.

36. H. Wakita, "Identification of Normalized Steady-State Vowels", 90th Meeting of the Acoustical Society of America, San Francisco, California, November 1975.

37. H. Wakita, "Estimation of Vocal Tract Length from Acoustic Data", 87th Meeting of the Acoustical Society of America, New York, New York, April 1974.

38. T. G. Stockham, Jr., et al., "Blind Deconvolution through Digital Signal Processing", *Proceedings of the IEEE*, Vol. 63, No. 4, April 1975.

APPENDIX I

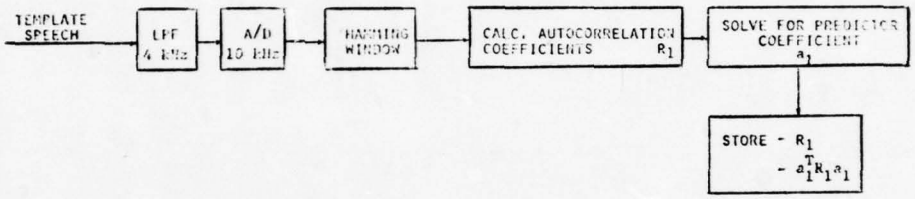
WORD-SPOTTING PROGRAM -- FORTRAN CODE

```
      DIMENSION IDIM(80).SIM(10),WI(256)
      DIMENSION AR(80,0/1),S(9216),IDATA(9216)
      DIMENSION SP(256),WS(256),R(14),A(12),B(14)
      DIMENSION RR(14,80,10),              RN(80,10)
      COMMON /VM/AL
      COMMON/IEOF/IEOF1
      COMMON /TS/ G,Q,RK,IDEV1
      INTEGER STATUS,ENDBLK,DO
      LOGICAL FLAG
      DATA NW,NW2,IDEV1,INIBLK,M/128,256,5,1000,12/
      DO = 0
      CALL DING(1)
      CALL CLKSTP
      CALL FREQST(10000.)
      CALL CLKGO
      CALL DING(1)
      TYPE 12,IDEV1
12    FORMAT(' IDEV1 = ', I2,'    ',$)
      CALL INNUM(IDEV1)
      TYPE 425,M
425   FORMAT(' M= ',I2,'    ',$)
      CALL INNUM(M)
      TYPE 426,ITESTM
426   FORMAT(' ITESTM= ',I2,'    ',$)
      CALL INNUM(ITESTM)
      DO 3 I=1,ITESTM
3     IDIM(I)=0
      NPTS = 256
      F = (3.1415926/(NPTS - 1))*2.
      DO 2 K=1,NPTS
2     WI(K)=.54-.46*COS((K-1)*F)
C             READ AND WINDOW SAMPLES OF SPEECH
      DO 99 KL=1,ITESTM
5     CALL OPEN(DO,NBLKS,ENDBLK,INIBLK,IDEV1)
      NBLK = INIBLK
10    CALL DSKIO(IDEV1,NBLK,2,NW,SP,STATUS)
      IF(STATUS) 51,52,53
51    TYPE 202
```

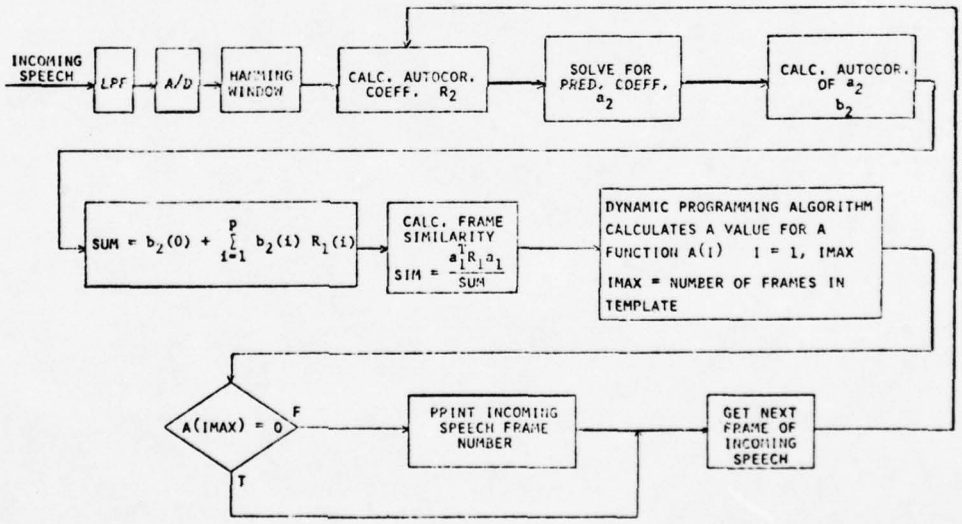
```
202     FORMAT(' HARDWARE')
        GO TO 203
53      TYPE 204
204     FORMAT( ' PARAMETER')
205     CONTINUE
        CALL SLEEP(10)
        GO TO 10
52      CONTINUE
        NBLK =NBLK+1
        IF(NBLK.GT.ENDBLK)GO TO 99
        CALL EXPNDO(NW2,SP)
203     DO 20 K=1,NPTS
20      WS(K) = SP(K)*WI(K)
C               CALCULATE AUTOCORRELATION COEFFICIENTS
        CALL CORR(WS,NPTS,R,M+1)
        RNORM = 1./R(1)
        R(1) = 1.
        DO 32 K=2,M+1
32      R(K) = R(K)*RNORM
C               SOLVE FOR PREDICTOR COEFFICIENTS
        CALL SOLVE(R,A,M)
50      IDIM(KL) = IDIM(KL) + 1
        IF(KL.EQ.ITESTM)CALL MMX(IDIM,80,IMIN,IMAX)
        RN(IDIM(KL),KL) = AL
        DO 60 K = 1,M+1
60      RR(K,IDIM(KL),KL) = R(K)
        GO TO 10
99      CONTINUE
        GO TO 500
100     CONTINUE
700     CALL OPEN(DO,NBLKS,ENDBLK,INIBLK,IDEV1)
        KT = 1
        DO 1100 L=1,80
        AR(L,0)=0.
 1100   AR(L,1)=0.
        NBLK = INIBLK
701     CALL DSKIO(IDEV1,NBLK,2,NW,SP,STATUS)
        IF(STATUS) 702,707,704
702     TYPE 703
703     FORMAT(' HARDWARE')
        GO TO 708
704     TYPE 705
705     FORMAT( ' PARAMETER')
706     CONTINUE
        CALL SLEEP(10)
        GO TO 701
707     CONTINUE
        NBLK =NBLK+1
```

```
      IF(NBLK.GT.ENDBLK)GO TO 500
      CALL EXPNDO(NW2,SP)
708   DO 709 K=1,NPTS
709   WS(K) = SP(K)*WI(K)
C               CALCULATE AUTOCORRELATION COEFFICIENTS
      CALL CORR(WS,NPTS,R,M+1)
      RNORM = 1./R(1)
      R(1) = 1.
      DO 710 K=2,M+1
710   R(K) = R(K)*RNORM
C               SOLVE FOR PREDICTOR COEFFICIENTS
      CALL SOLVE(R,A,M)
C               CALCULATE FRAME SIMILARITY ---SIM---
      CALL BAUTO(A,B,M)
      DO 111 KM=1,ITESTM
      SUM = B(1)
      DO 110 I = 1,M
110   SUM = SUM + B(I+1)*RR(I+1,1,KM)
      SIM(KM) = RN(1,KM)/SUM
111   CONTINUE
      CALL MMX(SIM,10,SMIN,SMAX)
      IF(SMAX.LT.Q.AND.KT.EQ.1)GO TO 701
      KT = KT + 1
      IF(SMAX.GE.Q) GO TO 1000
      X = 0.
      GO TO 1010
1000  X = SMAX
1010  CALL STEP(AR(1,0),X,RK,ST)
      AR(1,1) = AMAX1(X,ST)
      DO 120 K = 2,IMAX
      IF(AR(K,0).EQ.0..AND.AR(K-1,0).EQ.0..AND.AR(K-1,1).EQ.0.)
     1 GO TO 600
      DO 135 KN=1,ITESTM
      SUM =B(1)
      DO 125 I = 1,M
125   SUM = SUM + B(I+1)*RR(I+1,K,KN)
      SIM(KN) = RN(K,KN)/SUM
135   CONTINUE
      CALL MMX(SIM,10,SMIN,SMAX)
      CALL STEP(AR(K-1,1),SMAX,RK,A1)
      CALL STEP(AR(K-1,0),SMAX,1.,A2)
      CALL STEP(AR(K,0)SMAX,RK,A3)
      AR(K,1) = AMAX1(A1,A2,A3)
      GO TO 120
600   AR(K,1)=0.
120   CONTINUE
      DO 130 I = 1,80
130   AR(I,0) =AR(I,1)
      IFR = NBLK-1
```

```
        HIT = FLOAT(IFR)
        DO 7 I=1,ITESTM
        WIN=FLOAT(I)
        IF(AR(IDIM(I),1).NE.0.)GO TO 6
        GO TO 7
6       IF(IABS(IFR-IT).LT.20)GO TO 7
        TYPE 400,WIN,HIT,AR(IDIM(I),1)
        IT = IFR
        NB = IFR-36
        CALL DSKIO(IDEV1,NB,2,4608,S,STATUS)
        CALL EXPNDO(9216,S)
        CALL FIXAR(IDATA,S,9126)
513     CALL DMPBUF(IDATA,9216,FLAG)
        IF(FLAG) GO TO 513
7       CONTINUE
        GO TO 701
500     CALL DING(3)
        IT = 0
        TYPE 510,DO
510     FORMAT(' DO = ',I2,'    ',$)
        CALL INNUM(DO)
        IF(DO.NE.0) GO TO 700
400     FORMAT(8F12.2)
        END
```



ANALYSIS FOR REFERENCE TEMPLATE



ANALYSIS OF INCOMING SPEECH

Fig. A.1. Flow chart for Fortran listing.

APPENDIX II

PDP-11 PASSAGE READ BY SUBJECTS

A stack is a first-in last-out list. In a PDP-11 a stack is used automatically by program interrupts, subroutine calls, and trap instructions. When the processor is interrupted, the central processor status word and the program counter are saved (or "pushed") in the stack area, while the processor services the interrupting device. A return from interrupt instruction restores the interrupted program without software intervention. Interrupts are then automatically nested for fastest interrupt response and no program overhead.

The PDP-11 has an extremely flexible interrupt priority structure. The central processor recognizes interrupts on four separate lines. Many devices may be attached on each line with the device closest to the central processor given priority over the other devices on the same line. The hardware interrupt priority lines are interleaved by programmable central processor priority levels, thus allowing the running program to select the priority of allowable interrupts. Additional speed and power are added to the interrupt structure through the use of the PDP-11 fully vectored interrupt scheme.

When acknowledging an interrupt, the processor saves its program control and status word on the system push-down stack. Then, it

establishes a new status and program counter from a set of interrupt locations unique to each device. Thus, no device polling is required. The program running in response to an interrupt may itself be interrupted by a device with a higher priority; these nested interrupts are allowed to any level. Subroutine call instructions also use the stack for storage of the return address. Hence, reentrant subroutines may be easily written. Reentrant subroutines make it possible to do real-time programming without using large amounts of core memory.

APPENDIX III

FORD PASSAGE

President Ford's advisers are urging him to revamp his strategy -- travel less, project a more positive image, and beef up his campaign organization. Friends contend that Mr. Ford's trips have boosted Republican morale and filled the party's coffers, but they haven't improved Ford's poll ratings.

Raising money for Ford is turning out to be tougher than the White House expected. Strategists say that President Ford's unyielding stand so far on hot issues has turned off contributors.

Ford campaign chief, Howard Callaway, is catching flak. Insiders look for Donald Rumsfeld, top White House aide, to make more key political decisions. President Ford's team expects Ronald Reagan to go all out in early primaries, hoping to make Mr. Ford look weak.

For candidate Jerry Ford, money has suddenly become more of a problem than expected only a short time ago. Campaign cash hasn't rolled in and the Ford Committee may have to turn to federal matching funds. Even so, Mr. Ford will definitely enter six of the early State primaries next year.

President Ford, although not to the extent of his immediate predecessors, is showing annoyance at "inaccurate" reporting of White House activities.

ACKNOWLEDGMENTS

I wish to express my appreciation to the members of my Supervisory Committee for their help in this work. Special thanks are due to Dr. Craig K. Rushforth for his constant support, criticism, and guidance. He always made available what seemed like an unlimited amount of time for discussion and review of this work. Thanks are also due to Dr. Thomas G. Stockham, Jr., for suggesting the problem. I also thank Dennis Pulsipher and William Done for much assistance in programming and the use of the computer.

I must also acknowledge the continual support and cooperation of my wife, Anita, without which this work might not have been completed.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER UTEC-CSc-76-226	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* WORD RECOGNITION IN CONTINUOUS SPEECH USING LINEAR PREDICTION ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard Wesley Christiansen		8. CONTRACT OR GRANT NUMBER(s) DAHC15-73-C-0363
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Department University of Utah Salt Lake City, Utah 84112		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order #2477
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, Virginia 22209		12. REPORT DATE August 1976 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*
This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*
word spotting, error residual ratios, LPC, vocoder analysis, templates, word recognition.

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A promising method of automatic word recognition in continuous speech, recently designated as word spotting, has been demonstrated. The method uses error residual ratios from LPC (Linear Predictive Coding) vocoder analysis for waveform comparison and a dynamic programming procedure for time registration between the invoming speech and a template of the key word. Using a similarity threshold, the incoming speech is compared with several templates to account for variability in spectral shape. This system can work in real time using a real

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

(con't) 20. Abstract

time vocoder.

The multiple templates are used in such a way that a small number of templates, three or four, is made to look like several hundred or more. This is accomplished by dynamically constructing a composite template from parts of each single template as part of the processing of the incoming speech. Thus, a particular composite template is constructed for each word being recognized.

An accuracy of 99 percent with no false alarms was achieved using 205 key words, five different speakers, and approximately ten minutes of speech text. Performance in the presence of additive white gaussian noise of approximately 11 dB signal-to-noise ratio was 66 percent. When the speech was processed to account for the noise, results improved to 85 percent to 90 percent accuracy. Finally a digit recognition experiment was performed using over 1200 digits spoken by ten different people with a resultant accuracy of 97 percent.